LETTER COMBINATORICS: A THEORY OF COUNTING PROBLEM.

(Letter Marriage Method)

FRANK K. APPIAH[1]
UNIVERSITY OF CAMBRIDGE
KING'S COLLEGE
CAMBRIDGE, UK.

---

1 Dr. Frank Appiah hold studentships at the Bsc. (Bachelor in computer engineering, Kwame Nkrumah University of Science and Technology) in 2008, Msc.(software engineering, King's College London) in 2009/10 and 2 doctorates of philosophy (DPhil / PhD) in computer engineering from Kwame Nkrumah University of Science and Technology in 2013/14. He did his doctor of philosophy in computer science and engineering at King's College London in 2010/12 and postgraduate studies at the same place. He is a member of King's Alumni Group of Engineering, British Academy and Royal Society.

This is a teaching work in its first publication at the above institution. Any omitting or errors are mine only and can be communicate to when needed.

TEACHING POSTER

(Applied Mathematics

AND

Computer Science)

ABSTRACT

This teaches the theory of combinatorics of sentences or phrases or words called Letter Combinatorics (*LC*). A Letter Marriage Problem (*LMP*) set is used in the exploit of letter combinatorics. This example of letter combinatorics shows the calculation of *addition*, *multiplication* and *subtraction* principles of LMP.

1.INTRODUCTION

1.Letter Combinatorics has or must have the following requirements:

- A countable number of sentences or phrases.
- Counting the size of selected phrases for likely occurrence of subset equality of letters is called *count*.
- A sentence with the number of letters specified is called Π.
- The logical structure of arithmetic such as +, - and = should be applied.
- Discrete operations must include count, addition, subtraction and sizes.
- The sizes of selected phrases are enumerated.

•Proofs with the discrete operations on which the enumeration of sizes stops.

•There is a possibility of summation equal to Π.

A logical Structure is of the form:

$$L = \langle \ + | - \rangle$$
$$=, \sum_{s}^{n} \ , where n > 1$$
.

n is a sentence or phrase number and s under sum means sentential summation.

2.***<u>(Count Equality Principle) Definition</u>***: *The size of selected phrases for likely occurrence of subset equality of letters.*

An *r-combination* of *n* distinct objects is an unordered selection, or subset of *r* out of the *n* objects(letters). The fundamental skills of

combinatorial reasoning on letter combinatorics is simply a class of counting problems. Count Equality is a principle of solution of specific classes of counting problem with no arrangement but selection with repetition. The two main counting principles in the elements of classes of counting problem are addition and multiplication principles.

3*Addition Principle* states that if there are r_1 different objects in the first set, r_2 different objects in the second set, ..., and r_m different objects in the mth set, and if the different sets are disjoint, then the number of ways to select an object from one of the m sets is $r_1 + r_2 + r_3 + r_4 \dots + r_m$. On the other hand, *Multiplication Principle* supposes a procedure can be broken into successive (ordered) stages with r_1 different outcomes in the first stage, r_2 different

outcomes in the second stage, …, and r_m different outcomes in the mth stage. If the number of outcomes at each stage is independent of the choices in previous stages and if the composite outcomes are all distinct, the total procedure has

$r_1 \times r_2 \times r_3 \times r_4 \ \ldots \ \times r_m$ different composite outcomes.

4.A distribution problem is equivalent to an arrangement or selection problem with repetition. The focus on modeling distribution problems can be broken into sub-cases that can be counted in terms of simple permutation and combinations. The process of distributing r identical letter objects into n different letter objects is the selection equivalence of distribution.

Selection Equivalence of Letter Distribution[SELD] (Corollary 1): *The* d*istributions of identical letter objects(word) are equivalent to letter selections.*

Thus, there $C(r+n-1\ ,r) = \frac{(r+n-1\ ,r)!}{r!\,(n-1)!} \equiv SELD$

__ $C(n-r)$ ***- Combination (Corollary 2)***: *Letter combinations are a general counting method of unordered outcome.*

Distributing of distinct letter objects are equivalent to arrangements but letter combinations do not have that as a generally specialized distribution problem.

__ $n \equiv^{r}$ ***- Arrangement (Corollary 1):*** Letter

combinations are not a general distribution problem of distinct letter objects.

__ $n \equiv^{r}$ ***- Arrangement (Corollary 2):*** Letter combinations are not a general arrangement of ordered outcome.

5.***(Letter Count Problem) Theorem***: *Letter combinatorics is a counting problem.*

6.(***Letter Cut) Lemma***: *A selected phrase is by cutting a number of possible letters.*

7.Finally, it is a counting problem and a selected phrase is by cutting a number of possible letters.

8.***(Cutting Strategy) Proposition***: *A strategy of cutting the possible matches can continue with as*

many comparisons as needed.

9.***(Π-Sentence) Axiom***: *Π is a number of letters specified in a sentence.*

2.0 MARRIAGE PROBLEM (EXAMPLE)

(1) Damn it.

(2) What's wrong?

(3) It is a combination of 46 letters.

(4) Akua will not marry you.

(5) Pokua will not marry you.

10. *Partition of Integers (Definition): A partition of countable integer, count to be a collection of positive integers whose sum is N.*

11. For the Marriage Problem(MP) Example, the partition of integers are:

Damn it. (1)

$4 + 2 = N = 6.$

What's wrong (2)

$5 + 5 = N = 10.$

12. The collection or set of a sum and the list of integers of the partition is in increasing order. *MP = { 6, 10 }*, *MP* set is a set of marriage problem partitions of integers. The problem, MP 2 is equivalent to the number of integer solutions to

$$5e_5 + 5e_5 = 10e_5 = |5+5| = 10 \quad .$$

The generating function for the number(6: MP 2

case), is the ways that we can choose countable integers

$$\left(1+X^5+X^{10}+X^{15}\ldots\right)\left(1+X^5+X^{10}+X^{15}\ldots\right).$$

Generating functions are to handle constraints in selection and arrangement problems.

13. Combinatorial Enumeration of Letters

- $\sum_s N$ is the sentential summation for a sentence.

Letter Marriage Problem is a sentential summation operated by subtraction on minimum and maximum values given as LMPSet={6, 10, 14, 19, 20, 27}. LMPSet is union of RMPSet[2] *and sum(MP_6).*

•***Π = 46.***

•*Logical Structure*

$$L=\left(+,\sum_s^4,\sum_s^6\right)$$

•*Discrete Operation (Proof)*

2 Letter Combinatorics on Real Marriage Problem

•The Equality Principle on the Marriage Problem is

$$\sum_{s}^{L} = \sum_{s}^{4} + \sum_{s}^{6}$$
$$= 19 + 27$$
$$= 46. \qquad = \boldsymbol{\Pi}$$

$\boldsymbol{\Pi}$ stops on the enumeration of size 46 with the sentential summation for sentence 4 and sentential summation for sentence 6.

•*Discrete Operation (Addition)*

The Addition principle on the Letter Marriage Problem results:

$$\sum_{s}^{L_A} = \sum_{s}^{1} + \sum_{s}^{2} + \sum_{s}^{3} + \sum_{s}^{4} + \sum_{s}^{5} + \sum_{s}^{6}$$
$$= 6 + 10 + 14 + 19 + 20 + 27$$
$$= 96$$

.

•*Discrete Operation (Multiplication)*

The Multiplication principle on the Letter Marriage

Problem results:

$$\sum_{s}^{L_M} = \sum_{s}^{1} x \sum_{s}^{2} x \sum_{s}^{3} x \sum_{s}^{4} x \sum_{s}^{5} x \sum_{s}^{6}$$
$$= 6 \ x \ 10 \ x \ 14 \ x \ 19 \ x \ 20 \ x \ 27$$
$$= 8618400$$
.

•*Discrete Operation (Subtraction)*

The Subtraction principle of Letter Marriage

Problem results:

$$\sum_{s}^{L_S} = \sum_{s}^{6} - \sum_{s}^{5} - \sum_{s}^{4} - \sum_{s}^{3} - \sum_{s}^{2} - \sum_{s}^{1}$$
$$= 27 - 20 - 19 - 14 - 10 - 6$$
$$= -42$$
.

The discrete subtraction operation of LMP has equal value to the real marriage problem(RMP).

•Discrete Meta-Operation (Addition-Subtraction)

$\sum_{s}^{L_{AS}}$ is the meta-sentential summation.

$$\sum_{s}^{L_{AS}} = \left\{ \sum_{s}^{L_1}, \sum_{s_3}^{L_2}, \sum_{s_2}^{L_5}, \sum_{s_1}^{L_3}, \sum_{s}^{L_6}, \sum_{s_3}^{L_4} \right\}$$

This operation is based on the logical structure as shown below:

$$\left(+, \sum_{s}^{3}, \sum_{s}^{1} \right) = \sum \left(L_{5,} s \right).$$

$$\sum_{s}^{5} = \sum_{s}^{3} + \sum_{s}^{1}$$
$$= 14 + 6$$
$$= 20$$

The operation to determine the results of **sum**(L_6, s) has a logical structure:

$$L = \left(+ \mid -, \sum_{s}^{4}, \sum_{s}^{3}, \sum_{s}^{1} \right) = \mathbf{sum}(L_6, s).$$

$$\sum_{s}^{6} = \sum_{s}^{4} + \sum_{s}^{3} - \sum_{s}^{1}$$
$$= 19 + 14 - 6$$
$$= 27$$

The operation to determine the results of **sum**(L_4, s) has a logical structure:

$$L=\left(+ \mid - , \sum_{s}^{5} , \sum_{s}^{4} , \sum_{s}^{2} \right) = \textbf{sum}(L_4, s).$$

$$\sum_{s}^{4} = \left(\sum_{s}^{4} - \sum_{s}^{2} \right) + \left(\sum_{s}^{5} - \sum_{s}^{2} \right)$$
$$= (19 - 10) + (20 - 10)$$
$$= (9 + 10)$$
$$= 19$$

This operation, **sum**(L_3, s) is based on the logical structure shown below:

$$L=\left(- , \sum_{s}^{5} , \sum_{s}^{1} \right) = \textbf{sum}(L_3, s).$$

$$\sum_{s}^{3} = \sum_{s}^{5} + \sum_{s}^{1}$$
$$= 20 - 6$$
$$= 14$$

This operation, **sum**(L_1, s) is based on the logical

structure shown below:

$$L = \left(- \;,\; \sum_{s}^{5} ,\; \sum_{s}^{3} \right) = \mathbf{sum}(L_1, s).$$

$$\begin{aligned} \sum_{s}^{1} = \sum_{s}^{5} - \sum_{s}^{3} \\ = 20 - 14 \\ = 6 \end{aligned}$$

This meta-operation, **sum**(L_2, s) is based on the logical structure shown below:

$$L = \left(- \;,\; \sum_{s}^{5} ,\; \sum_{s}^{2} \right) = \mathbf{sum}(L_2, s).$$

$$\begin{aligned} \sum_{s}^{2} = \sum_{s}^{5} - \sum_{s}^{2} \\ = 20 - 10 \\ = 10 \end{aligned}$$

This is the complete meta-operations of the letter marriage problem. The purpose is to generate the counts

of the Letter Marriage Problem from the same counts of sentential summations of sentence. This is all that meta-operation is about. A discrete operation about another discrete operation is a discrete meta-operation. It seems that there is only one discrete operation in the logical structure for some cases. This is not the case, the count of sentence or size of letters/word is all based on additive operation. **sum**(L_4, s) is a perfect discrete meta-operation that uses both subtraction and addition operations.
The sentence for 4 states:

(4) Akua will not marry you.

3.0 SUMMARY

This is letter combinatorics on a letter marriage problem. This teaches what is all about letter combinatorics of words, sentences or phrases. The Equality Principle on Marriage Problems is calcu-

lated and resulted as 46 for Marriage Problems. The discrete subtraction operation of LMP is the same to the real marriage problem (RMP) is the first research finding. The meta-operation of LMP is the second research finding based on the determination of all counts of LC on LMPset. The discrete operations can be generalized by the equations:

- $a+b$
- $a-b$
- $a+b-c$ and
- $(a-b)+(c-b)$.

The discrete set operations can be generalized by the equations:

- a_n-b_n
- a_n+b_n
- $a_n+b_n-c_n$ and
- $(a_n-b_n)+(c_n-b_n)$.

Appendix : Insertion Object Formula-OpenOffice Writer.

Page 5: Logical Structure

$$L= \langle +|-\rangle =, \sum_{s}^{n} ,where n>1$$

```
size 12{alignc { stack {
ital "L="~ langle ~+` mline  - ` rangle  {} #
` ital "=,"~ Sum cSub { size 8{s} }  cSup { size 8{n}
}  {~}  ital ",where n>"1 {}
} } } {}
```

Page 6: Addition Principles r_1

```
size 12{r rSub { size 8{1} } } {}
```

Page 7: Addition Principles $r_1 + r_2 + r_3 + r_4 \ldots + r_m$

size 12{r rSub { size 8{1} } `+`r rSub { size 8{2} }
`+`r rSub { size 8{3} } `+`r rSub { size 8{4} } dotslow `+`r
rSub { size 8{m} } } {}

Page 8: Multiplication Principles

$r_1 \; x \; r_2 \; x \; r_3 \; x \; r_4 \; \dots \; x \; r_m$

size 9{r rSub { size 8{1} } `x`r rSub { size 8{2} } `x`r rSub { size 8{3} } `x`r rSub { size 8{4} } ~ dotslow ~x`r rSub { size 8{m} } } {}

Page 8: SELD $C\left(r+n-1 \;, r \right) = \frac{\left(r+n-1 \;, r \right)!}{r! \left(n-1 \right)!} \equiv \; SELD$

size 9{C left (` ital "r+n" - 1` ital ",r" right)`=` { { left (` ital "r+n" - 1` ital ",r" right)!} over {` ital "r!"` left (`n - 1` right)!} } ~ equiv ~ ital "SELD"} {}

Page 9: $n \equiv^{r}$

size 9{n equiv rSup { size 8{r} } `} {}

Page 12: $5e_5 \; + \; 5e_5 = \; 10e_5 = \left(5+5 \right) = 10$

size 9{5e rSub { size 8{5} } ~+~5e rSub { size 8{5} }`=~10e_5 `=~ left (5+5 right)`=`10} {}

Page 12: $\left(1 + X^5 + X^{10} + X^{15} \dots \right) \; \left(1 + X^5 + X^{10} + X^{15} \dots \right)$

size 9{ left (1 ital "+X" rSup { size 8{5} } ital "+X" rSup { size 8{10} } ital "+X" rSup { size 8{15} } ` dotslow right )~ left (1 ital "+X" rSup { size 8{5} } ital "+X" rSup { size 8{10} } ital "+X" rSup { size 8{ ital "15"} } ` dotslow right)} {}

Page 13: $\sum_{s} N$

size 9{ Sum cSub { size 8{s} } {`} N} {}

$$L= \left(+, \sum_{s}^{4}, \sum_{s}^{6} \right)$$

size 9{ ital "L="~ left (~ ital "+,"~ Sum cSub { size 8{s} } cSup { size 8{4} } {,} ~ Sum cSub { size 8{s} } cSup { size 8{6} } {~} right)} {}

Page 14: Addition Operation

$$\begin{aligned} \sum_{s}^{L_A} &= \sum_{s}^{1} + \sum_{s}^{2} + \sum_{s}^{3} + \sum_{s}^{4} + \sum_{s}^{5} + \sum_{s}^{6} \\ &= 6+ \ 10+ \ 14+ \ 19+ \ 20+ \ 27 \\ &= 96 \end{aligned}$$

size 9{alignc { stack {

Sum cSub { size 8{s} } cSup { size 8{L rSub {A} } } {`} =~ Sum cSub { size 8{s} } cSup { size 8{1} } {`} +~ Sum cSub { size 8{s} } cSup { size 8{2} } {`} +~ Sum cSub { size 8{s} } cSup { size 8{3} } {`} +~ Sum cSub { size 8{s} } cSup { size 8{4} } {`} +~ Sum cSub { size 8{s} } cSup { size 8{5} } {`} +~ Sum cSub { size 8{s} } cSup { size 8{6} } {~} {} #

```
`=`6+~"10"+~"14"+~"19"+~"20"+~"27" {} #
`=`"96" {}
} } } {}
```

Page 14: Multiplication Operation

$$\sum_{s}^{L_M} = \sum_{s}^{1} x \sum_{s}^{2} x \sum_{s}^{3} x \sum_{s}^{4} x \sum_{s}^{5} x \sum_{s}^{6}$$
$$= 6 \quad x \quad 10 \quad x \quad 14 \quad x \quad 19 \quad x \quad 20 \quad x \quad 27$$
$$= 8618400$$

size 9{alignc { stack {

Sum cSub { size 8{s} } cSup { size 8{L rSub {M} } } {`} =~ Sum cSub { size 8{s} } cSup { size 8{1} } {`} x~ Sum cSub { size 8{s} } cSup { size 8{2} } {`} x~ Sum cSub { size 8{s} } cSup { size 8{3} } {`} x~ Sum cSub { size 8{s} } cSup { size 8{4} } {`} x~ Sum cSub { size 8{s} } cSup { size 8{5} } {`} x~ Sum cSub { size 8{s} } cSup { size 8{6} } {~} {} #
`=`6~x~"10"~x~"14"~x~"19"~x~"20"~x~"27" {} #
`=`"8618400" {}
} } } {}

Page 15: Subtraction Operation

$$\sum_{s}^{L_S} = \sum_{s}^{6} - \sum_{s}^{5} - \sum_{s}^{4} - \sum_{s}^{3} - \sum_{s}^{2} - \sum_{s}^{1}$$
$$= 27 - \quad 20 - \quad 19 - \quad 14 - \quad 10 - \quad 6$$
$$= -42$$

size 9{alignc { stack {

Sum cSub { size 8{s} } cSup { size 8{L rSub {S} } }

{`} =~ Sum cSub { size 8{s} } cSup { size 8{6} } {`} - ~ Sum cSub { size 8{s} } cSup { size 8{5} } {`} - ~ Sum cSub { size 8{s} } cSup { size 8{4} } {`} - ~ Sum cSub { size 8{s} } cSup { size 8{3} } {`} - ~ Sum cSub { size 8{s} } cSup { size 8{2} } {`} - ~ Sum cSub { size 8{s} } cSup { size 8{1} } {~} {} #
`=`"27" - ~"20" - ~"19" - ~"14" - ~"10" - ~6 {} #
`=` - "42" {}
} } } {}

Page 16: Meta-Discrete Operation

$$\left(+, \sum_{s}^{3}, \sum_{s}^{1} \right) = \sum \left(L_{5,} s \right)$$

size 9{~ left (~ ital "+,"~ Sum cSub { size 8{s} } cSup { size 8{3} } {,} ~ Sum cSub {size 8{s} } cSup { size 8{1} } {~} right)= Sum { left (L rSub { size 8{"5,"} } s right)} } {}

$$\begin{aligned} \sum_{s}^{4} = \left(\sum_{s}^{4} - \sum_{s}^{2} \right) + \left(\sum_{s}^{5} - \sum_{s}^{2} \right) \\ = (19 - 10) + (20 - 10) \\ = (9 + 10) \\ = 19 \end{aligned}$$

size 9{alignc { stack {
{} #
Sum cSub { size 8{s} } cSup { size 8{4} } {`} = left (` Sum cSub { size 8{s} } cSup { size 8{4} } {`} - ~ Sum cSub { size 8{s} } cSup { size 8{2} } {`} right)+ left (` Sum cSub { size 8{s} } cSup { size 8{5} } {`} ~ - Sum cSub { size 8{s} } cSup { size 8{2} } {`} right) {} #
`=` \("19"~ - ~"10" \) ~+~ \("20"~ - ~"10" \) {} #

```
`=` \( 9~+~"10" \)  {} #
`=`"19" {}
} } } {}
```

Page 20: General Discrete Operation

$a+b$

```
size 9{ ital "a+b"} {}
```

$a-b$

```
size 9{a - b} {}
```

$a+b-c$

```
size 9{ ital "a+b" - c} {}
```

$\left(a-b\right)+\left(c-b\right)$.

```
size 9{(a -  ital "b)+(c" - b)} {}
```

Page 20: General Discrete Set Operation

a_n-b_n

```
size 9{a_n - b_n} {}
```

a_n+b_n

```
size 9{ ital a_n+b_n} {}
```

$$a_n + b_n - c_n$$

size 9{ ital a_n+b_n - c_n} {}

$$\left(a_n - b_n\right) + \left(c_n - b_n\right)$$

size 9{(a_n - ital b_n)+(c_n - b_n)} {}

Bibliography

1. *Alan Tucker(2012), Applied Combinatorics, Wiley, 6th Edition, ISBN:1118210115.*